

AMENDMENTS TO THE CLAIMS

Please amend claims 1-2, 4-6, 9-14, 16, 18-21, and 23-25 as follows:

1. (Currently Amended) A method in a computer system for associating identifiers an identifier with virtual processes a virtual process, the virtual process comprising a plurality of processes, the method comprising:

~~for each virtual process, starting a separate first process;~~  
associating ~~each~~ the first process with a ~~separate~~ virtual process identifier;  
~~for each virtual process, originating, from the first process, a second process~~  
~~additional processes included in the virtual process from the first process;~~  
and  
~~associating all processes that originate from each first process~~ the second process  
with the ~~associated~~ virtual process identifier.

2. (Currently Amended) The method of claim 1 wherein:  
each the virtual process comprises a virtual private server, the virtual private  
server comprising a plurality of processes that together provide the  
functionality of a dedicated server application program.

3. (Original) The method of claim 1 wherein:  
the first process comprises a system initialization process.

4. (Currently Amended) The method of claim 1 wherein associating a the first  
process further comprises:

storing an entry in a data structure in computer memory, the entry comprising a  
the virtual process identifier and a process identification number of the  
first process.

5. (Currently Amended) The method of claim 1 further comprising:  
intercepting ~~system calls that create processes~~ a system call that creates a process;  
and  
associating a the process being created with the virtual process identifier of a  
process that made the system call.

6. (Currently Amended) The method of claim 5 further comprising:  
storing object code comprising instructions to associate the process being created  
with the virtual process identifier of the process that made the system call;  
and  
wherein intercepting comprises replacing a pointer to the system call with a  
pointer to the stored object code, such that calling making the system call  
causes the object code to execute.

7. (Original) The method of claim 6 wherein storing object code comprises inserting the  
object code into the operating system.

8. (Original) The method of claim 7 wherein:  
inserting object code into the operating system comprises loading a module into a  
running operating system kernel, the module comprising the object code.

9. (Currently Amended) The method of claim 1 further comprising:

loading, by a modified loader program, the first process into computer memory by

a modified loader program;

starting, by the modified loader program, the first process by the modified loader

program; and

storing, by the modified loader program, an entry in a data structure in computer  
memory, the entry comprising a the virtual process identifier and a process  
identification number of the first process.

10. (Currently Amended) The method of claim 1 further comprising:

starting, by a virtual process manager program, the first process by a virtual

process manager program; and

storing, by the manager program, an entry in a data structure in computer  
memory, the entry comprising a the virtual process identifier and a process  
identification number of the first process.

11. (Currently Amended) A computer program product for associating identifiers an  
identifier with virtual processes a virtual process, the virtual process comprising a plurality of  
processes, the computer program product comprising:

program code for starting, for each virtual process, a separate first process;

program code for associating each the first process with a separate virtual process  
identifier;

program code for originating, from each virtual process, from the first process, a  
second process additional processes included in the virtual process;

program code for associating ~~all processes that originate from each first process~~  
the second process with the associated virtual process identifier; and  
a computer readable medium on which the program codes are stored.

12. (Currently Amended) The computer program product of claim 11 further  
comprising:

program code for storing an entry in a data structure in computer memory, the  
entry comprising a the virtual process identifier and a process  
identification number of the first process.

13. (Currently Amended) The computer program product of claim 11 further  
comprising:

program code for intercepting ~~system calls that create processes~~ a system call that  
creates a process; and  
program code for associating a the process being created with the virtual process  
identifier of a process that made the system call.

14. (Currently Amended) The computer program product of claim 11 further  
comprising:

program code comprising instructions to associate the process being created with  
the virtual process identifier of the process that made the system call; and  
program code for replacing a pointer to the system call with a pointer to the  
instructions to associate, such that ~~calling~~ making the system call causes  
the instructions to associate to execute.

15. (Original) The computer program product of claim 14 further comprising:

program code for storing the instructions to associate in the operating system by loading a module into a running operating system kernel, the module comprising the instructions to associate.

16. (Currently Amended) The computer program product of claim 11 further comprising:

program code for starting the first process; and  
program code for storing an entry in a data structure in computer memory, the entry comprising a the virtual process identifier and a process identification number of the first process.

17. (Original) The computer program product of claim 16 further comprising:

program code for loading the first process into computer memory.

18. (Currently Amended) A computer system for associating ~~identifiers~~ an identifier with ~~virtual processes~~ a virtual process, the virtual process comprising a plurality of processes, the computer system comprising:

an initiation module, for starting a ~~separate first process for each virtual process~~;  
an origination module, coupled to the initiation module, for originating, from each ~~virtual process~~ the first process, additional processes a second process ~~included in the virtual process~~; and  
an association module, coupled to the origination module, for associating each the first process with a ~~separate~~ virtual process identifier, and for associating

~~all processes that originate from each first process~~ the second process with the associated virtual process identifier.

19. (Currently Amended) The system of claim 18 further comprising:  
a storage module, coupled to the association module, for storing an entry in a data structure in computer memory, the entry comprising a the virtual process identifier and a process identification number of the first process.

20. (Currently Amended) The system of claim 18 further comprising:  
an interception module, coupled to the association module, for intercepting ~~system calls that create processes~~ a system call that creates a process; and wherein  
the association module is further for associating a the process being created with the virtual process identifier of a process that made the system call.

21. (Currently Amended) The system of claim 20 further comprising:  
a storage module, coupled to the interception module, for storing object code comprising instructions to associate the process being created with the virtual process identifier of the process that made the system call; and wherein  
the interception module is further for replacing a pointer to the system call with a pointer to the stored object code, such that ~~calling~~ making the system call causes the object code to execute.

22. (Original) The system of claim 21 further comprising:

coupled to the storage module, an insertion module, for inserting the instructions to associate in the operating system by loading a module into a running operating system kernel.

23. (Currently Amended) A method in a computer system for associating identifiers an identifier with virtual processes a virtual process, the virtual process comprising a plurality of processes, the method comprising:

starting each the virtual process by executing a ~~separate,~~ system initialization process;  
associating each the system initialization process with a virtual process identifier;  
associating ~~all processes~~ a second process originating from each the system initialization process with the virtual process identifier ~~associated with the system initialization process;~~ and wherein  
a the virtual process is comprised of a ~~single~~ system initialization process and ~~all processes~~ a second process originating therefrom.

24. (Currently Amended) A method in a computer system for associating identifiers an identifier with virtual processes a virtual process, the virtual process comprising a plurality of processes, the method comprising:

associating initialization processes with respective virtual process identifiers; and  
in response to a process being created from one of the initialization processes,  
associating the process created by the initialization process with the virtual process identifier associated with the initialization process.

25. (Currently Amended) A method in a computer system for associating ~~identifiers~~  
~~an identifier with virtual processes~~ a virtual process, the virtual process comprising a plurality of  
processes, the method comprising:

loading, by a modified loader program, ~~each~~ a process that is a part of a the virtual  
process into computer memory ~~by a modified loader program~~; and  
storing, by the modified loader program, an entry in a data structure in computer  
memory, the entry comprising a the virtual process identifier and a process  
identification number of the loaded process.